# HOW TO RECOGNIZE YOU HAVE A SYSTEMS PROBLEM

As a senior manager, has this ever happened to you?

A senior executive finds “herself” faced with a product that is incurring major cost overruns or schedule delays. When she asks what the problem is, she’s told, “The requirements keep changing.” Or perhaps she hears, “A system component has to be significantly redesigned.”

These responses could be indications that there is a problem with product quality, choice of technology, or manufacturability difficulties. But there’s just as much chance that the culprit is a system problem.

To determine whether you do have a system problem, you need to perform three tests. These tests will show if delays and cost overruns are arising for one or two reasons. The first could be because the components of the product don’t interface with each other as a system. This is analogous to putting together a child’s bike. If the parts in the box don’t fit, then you won’t have something that looks like a bike.

The second could be due to incomplete requirements that went into building the components, which would result in a system not functioning as expected. In the bike example, even if the parts do fit, the bike may not work as intended. For instance, the bike doesn’t stop properly because the brakes don’t have proper contact with the wheels.

Test #1: Are the complaints of changing requirements primarily coming from the component builders? If so, the team that is responsible for the system design has not finished their work (completion of the design or integrity of the design). As with the bike analogy, the team is continuing to finalize the design of the bicycle.  And as it does, the team realizes that the gears ordered from the component builder will make it too difficult to pedal. Trying to get it right, the technical team changes its requirements to the gear manufacturer.

Test #2: Are the cost overruns or schedule delays caused by the system not behaving as expected during system test-and-integration activities? If so, your technical team didn't have the time to ensure the interfaces were fully defined and properly described to the component builders.  You might have components that fit together, but won't function as intended. As with the bike, you finish the assembly and discover that a rider can only pedal backward, due to the incorrect interface between the crank, the gear and the brake.

Test #3: Are the cost overruns and schedule delays arising because the user or investor is reporting that the system is not interacting with other systems as expected, and so expensive and time-consuming component fixes are needed to generate user or investor acceptance? The bike runs fine, but accessories, such as a standard aftermarket odometer, do not function properly.

If your answer is "yes" to any of the preceding questions, then your delays and cost overruns are probably the result of a system problem, and the root of the issue is the components were acquired before the system design was completed. The component builders have only a portion of the critical information they need to build proper components that will work within the system.  So the built components either will not interface with each other, or will not have the integrated capabilities that would allow the system to function as it should.  This will result in components and integration activities having to be reworked, often at great cost.

The risks associated with a system problem can be reduced if certain steps are taken before the project begins. If the senior executive is experiencing delays and cost overruns, she should ask herself: are there sufficient numbers of qualified system engineers, early enough, on the project?  And equally important – do the systems engineers have the time to fully complete the design before the components are acquired?  If not, the organization's programs will continually face delays and overruns.